Bachelor's thesis

Professional Sales & Information Technology

NMYYNS12 & NTIETS13H

2017

Roosa Saarikivi

SAP SCREEN PERSONAS CONCEPT DESIGN AND IMPLEMENTATION

– Case Wärtsilä

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Professional Sales & Information Technology

2017 | 48

Roosa Saarikivi

SAP SCREEN PERSONAS CONCEPT DESIGN AND IMPLEMENTATION

- Case Wärtsilä

The thesis was carried out at Wärtilä's Marine Solutions business unit as a user interface study. The SAP ERP system used by the company has been partially perceived as difficult and complicated. New business processes cause difficulties and the heavy user interface reduces efficiency of both new and old SAP users.

The aim of the study was to find out whether the SAP Screen Personas user interface can be applied cost-effectively to customisation-demanding business processes. Screen Personas is an SAP add-on that can be used for simplifying processes and creating customised user screens. A new user interface for the warranty handling process was developed, and the opportunities and weaknesses of Screen Personas were evaluated.

The research was implemented in a project-oriented manner and it included project management, budgeting, reporting, cost accounting and profitability analysis, in addition to the user interface development. The focus of the research was on the reporting of the development and results as well as on the evaluation of the benefits. The study did not consider the non-measurable benefits.

As a result of the thesis work, a simplified user interface was created which enables the warranty cases to be handled more efficiently in the future. The new interface enhanced the users' time usage, improved data quality and reduced errors. The results of this study can be applied to new process developments on the Screen Personas platform.

KEYWORDS:

usability, cost savings, 5S, SAP, Screen Personas, concept design, customisation

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Myyntityö & Tietotekniikka

2017 | 48

Roosa Saarikivi

SAP SCREEN PERSONAS -KONSEPTIN SUUNNITTELU JA TOTEUTUS

- Case Wärtsilä

Opinnäytetyö toteutettiin Wärtsilä Oy:n Marine Solutions -yksikössä käyttöliittymätutkimuksena. Yrityksen käytössä oleva SAP ERP -järjestelmä on osittain koettu hankalaksi ja monimutkaiseksi. Uudet bisnesprosessit aiheuttavat vaikeuksia ja raskas käyttöliittymä alentaa tehokkuutta sekä uusilla että vanhoilla SAP:n käyttäjillä.

Tutkimuksen tavoitteena oli selvittää, voiko SAP Screen Personas -käyttöliittymää soveltaa kustannustehokkaasti kustomointia vaativiin bisnesprosesseihin. Screen Personas on SAP:n lisäosa, jonka avulla voidaan yksinkertaistaa prosesseja ja luoda räätälöityjä käyttäjänäkymiä. Tutkimuksessa kehitettiin uusi käyttöliittymä takuuprosessille, ja samalla arvioitiin Screen Personasin mahdollisuuksia ja heikkouksia.

Tutkimus suoritettiin projektiluontoisesti, ja siihen sisältyi käyttöliittymäkehityksen lisäksi projektinhallinta, budjetointi, raportointi, kustannuslaskenta ja kannattavuuslaskenta. Tutkimuksen painopiste oli kehityksen ja tulosten raportoinnissa sekä hyödyn arvioinnissa. Tutkimus ei ottanut kantaa ei-mitattavissa oleviin hyötyihin.

Työn tuloksena syntyi yksinkertaistettu käyttöliittymä, jonka ansiosta takuutapauksia voidaan vastedes hoitaa tehokkaammin. Uusi käyttöliittymä tehosti käyttäjien ajankäyttöä, paransi tiedon laatua ja vähensi virheitä. Tutkimuksen tuloksia voidaan tulevaisuudessa soveltaa uusiin prosessikehityksiin Screen Personas -alustalla.

ASIASANAT:

käytettävyys, kustannussäästöt, 5S, SAP, Screen Personas, konseptisuunnittelu, kustomointi

CONTENTS

EQUATIONS

PICTURES

TABLES

LIST OF ABBREVIATIONS

ARR	Accounting Rate of Return measures the average annual rate of return of an investment
ERP	Enterprise Resource Planning is a real-time technology to manage core business processes
IRR	Internal Rate of Return computes the exact rate of return that an investment composes
JavaScript	Script language used in web environment
NPV	Net Present Value measures the net profit of an investment
PV	Present Value measures the present day value of an investment
SAP	SAP is a provider of enterprise software such as enterprise resource planning systems
Screen Personas	SAP add-on used to customize standard SAP screens and automate process steps

1 INTRODUCTION

Today IT (Information Technology) solutions are highly used to manage the enterprises' resource planning. The tools consist of all core business processes from product planning to service delivery. The commissioner of this thesis, Wärtsilä Ltd, uses SAP as the ERP tool to manage day-to-day business in real-time.

At Wärtsilä, it has been discovered that some employees find it difficult and time-consuming to use SAP as the enterprise resource planning tool. Many various processes are processed with same screens and transactions, which means that one business process can have over thousand fields displayed but of which only 100 are relevant to the business process. This means that the user must enter several transactions, tabs and buttons to go through the process. Therefore, the process becomes time-consuming, the information the users need is not easily accessible, and there is much room for data-entry errors.

To solve the usability issues and enhance the user satisfaction, Wärtsilä has considered to take in use SAP add-on software. One of the add-on software is Screen Personas that can be used to customize standard SAP screens to a specific group of users. In Screen Personas, SAP screens can be combined and standardized to give the users a more consistent look and scripts can be written to automate process steps.

The objective of this thesis was to define whether the usability needs of Wärtsilä's SAP users can be met with Screen Personas in a cost-efficient way. This study analyses the capabilities and benefits of Screen Personas for Wärtsilä's use and measures the profitability of using Screen Personas in specific business processes. The study finds answers to the questions: Can Screen Personas ease the work of the users? Can it enhance user experience and satisfaction and in that way improve user productivity? When will it pay itself back? What is the added-value?

The thesis is divided into four main sections: usability challenges, economy and profitability, case and conclusion. The usability challenges chapter takes a look at the usability issues found, the reasons why they occur and the ways to fix them. The economy and profitability chapter answers to the question why the usability issues should be fixed in terms of efficiency and productivity. The following chapter describes the case

implemented and its outcomes. The final chapter sums up the findings and presents the conclusion of the study.

2 USABILITY CHALLENGES

SAP applications are often designed with one-size-fits-all mentality which means processes are made consistent and standardized by the vendor in a way that many of the organization's different processes use the same transactions and screens, but different partitions within them. This results in users seeing multiple tabs and fields they do not need to see and in users' excessive navigation in the system. In some cases, immoderate access to information may lead to errors such as data entry errors that require even more time from the user to perform a specific task due to mistakes.

To build business processes uniquely in a way that each process has its own transactions and screens would be extremely expensive as there are hundreds of different processes performed in SAP and most of the similar processes are performed differently across plants and organizations. Finally, it would require that nearly each user would have their own SAP screens. In addition, building such individual screens for thousands of employees requires considerable resources. The creation of a simple user interface can easily take over 100 hours of work with a cost starting at 5 000 euros. The cost of building thousands of unique user interfaces would then cost millions. (Yalantis 2017.)

Therefore it is crucial to identify if there are similarities within the processes and the most serious usability issues the users face, in which business processes, in which transactions or screens and how many times. This chapter first studies the user: What is the user like? What is the user goal? What is the need of the user? After the user and the user actions are familiar, the reasons why the user needs are not fulfilled and why the user tasks are not effectively performed are examined. Finally, when the problems and challenges are identified, this chapter discusses how these issues can be fixed.

2.1 User, user goal and user task

A user is the person who interacts with the product. Sometimes the user is also the customer but many times, such as in software acquisitions, the user and the customer are not the same person. The difference between a user and a customer is that a user is the person who uses the product or service and a customer is the person who makes the purchase decision of the product or service. (Jokela 2010, 14.)

To make the division even more detailed, every user and every customer is unique and individual. In application design, it is very challenging to take each individual into account. First of all, each user has their own preferences, such as how they prefer to perform a task and in which order, what information they want to see and what visual look they prefer. Secondly, each customer has their own preferences which may vary significantly from the users in a case the customer and the user are different persons. In these cases when they are different persons, a customer bases its preferences money-wise by balancing between costs and benefits. A customer, who is not the end-user of the product or service, most likely chooses the option that is cheap and may put very little value on the actual user preferences.

To avoid ending up with a situation where the users dislike the application and are not willing to use it, it is crucial to listen to the users at all times of the design and acquisition process. Even though all users are unique and they have their own preferences, it is possible to build solutions that are desirable to at least most of the users. The first step is to categorise users into user groups, such as employees and superiors. These groups can still be defined in even smaller groups, for example, based on the users' level of experience, job roles, tasks, knowledge, skill, experience, education, habits and preferences. (Jokela 2010, 33-35.)

Categorising users based on their job roles is generally implemented in SAP. Employees who handle warranty claims as their job use all the same transactions and screens no matter if the material damaged is completely different or the information needed to input is different. When the number of warranty handlers is big, there could be a more specific screen or screens built to a specified group of users, for example, based on the plant they work on. Then, each user would have a more desirable workspace.

To understand what the users really want, it is important to know their goals and tasks: What is the user trying to achieve? What does the user really do? The user goal is the aim that the user strives to achieve. The user tasks are ways to achieve the goals. (UX 2013.) For example, the goal of the user can be to send a new part to a customer to replace a damaged part. The tasks to accomplish the goal can include receiving the customer claim, picking a new part from the warehouse and delivering it to the customer. The description of the user endeavour should contain, in addition to goals and tasks, frequency, duration of the performance and allocation between human and technological resources (Jokela 2010, 38).

2.2 Usability issues

From time to time the needs of the users are not fulfilled. It can be seen as users' frustration, dissatisfaction and lack of motivation. When users are unwilling to use software, the reason for dissatisfaction usually lies on the usability of the software. Usability problems persist in organizations that put little value on user experience. The ignorance of user needs and internal project communication when designing or acquiring software are the biggest reasons to usability problems. (Hyysalo 2009, 13.)

Usability problems are caused by various, simple and complex reasons. In many cases, the same factors that cause the problems are the same that complicate the problems to be fixed. Lack of resources is one of the common reasons to usability problems. When an organization lacks skills, time, money, or other resources to build or maintain a system, it usually also ends up in hindering the usability fixes. Another reason is technical limitations that can lead to usability problems and limit solutions. Limitations may be caused by insufficient technical knowledge or vendor software. When using vendor software, changes and improvements can be difficult or sometimes impossible to implement. (Ross 2011.)

Organizational culture, communication issues and complex needs are other reasons to usability problems. Some organizations may find it difficult and costly to improve the existing system so they prefer leaving the system as it is, poor and complex. In addition, training issues can override poor usability. Problems occurred during the training of a system can be seen in a way that the problem is in the user and not in the software. Other communication issues include weakly expressed and explained recommendations, misunderstandings and misinterpretations. Usability recommendations that are presented in a paper or as a presentation can leave question marks and be misunderstood. Also, as there are usually several people involved in the development process, the usability preferences may change along the way if, for example, the developer has misinterpreted something, then the result will not be the agreed one. (Ross 2011.)

The usability problems also originate from inaccurate specification of user needs. People who interview actual end-users about their preferences or observe their actions may understand the user needs falsely or make false conclusions. In addition, sometimes the designers may base the specifications of user needs on how they would themselves

work or how they would believe users to work in such situations. The designers can have difficulties in understanding normal users' problems. (Saariluoma 2004, 29.)

In addition, even though the user needs have been specified correctly, the reason for usability issues can lie in testing of the software. Many times the programmers and the people, who have been involved in the project since the beginning, do the testing. After the testers are satisfied, the software is launched and the result may not respond to the desires of the user. (Filenius 2015, 195.)

When the technology used distracts or prevents the users from achieving their goals, they typically end up avoiding using the technology partially or entirely. Methods and ways applied for technical workarounds are, for example, paper notes that are used instead of writing the information directly into the system and instruction notes about how some systems function. Also, ignoring some of the process steps is a common workaround. (Hyysalo 2009, 47.)

2.3 Fixing usability issues

Finally, when the problems and issues are identified, the ways how these issues can be fixed must be studied. So far this chapter has examined what the usability problems are like, how they persist and why they persist. To not end up in an endless user dissatisfaction loop, this chapter also defines what good usability is and what kind of ways and means there are to fix the usability problems.

Good usability means an understandable, easy, comprehensive and aesthetically pleasing application. It is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. (Jokela 2010, 18.) The user must easily deduce what can be done with the application. An application with good usability provides all functions and information the user needs to handle the task for which the app is intended. The use performance is then efficient and flawless with a minimal risk for user mistakes. (Wiio 2004, 29-31.)

To ensure good usability, the end-users should be at the center of the design. Concepts related to it include human-oriented design and user-centric design. The aim of these concepts is to ensure precise and effective user interface design and to help the developers understand how the users work and in what kind of usage situations. (Saariluoma 2004, 31-32.) The reason why both of these concepts are taken into account

in this study is that human-oriented design focuses on human actions, which means that human, not the application, is in the center of development. On the contrary, the user-centric design brings a more in-depth aspect inside human-oriented design. Instead of paying attention to any of the humans involved in the design process, such as developer, project manager, project owner or user, the focus is entirely on the user. When combining the concepts of human-oriented and user-centric design, the user is given a much wider perspective in the design process.

Human-centered design starts with identifying the need and understanding and specifying the context of use. The need should answer to question: what is demanded? When the need is specified, the aspects related to the context of use should also be defined: what are the user tasks and goals? What is the user about to accomplish? How often are the tasks done? How critical are they to errors? How long time does it take to perform the tasks? In what kind of pressure are they done? The answers are followed by the specification of user requirements, creation of design solutions and evaluation of the solutions. (Jokela 2010, 26, 37.)

The following subchapters address in more detail how the need and context of use are specified and how the process flows towards a user-centric solution.

2.3.1 User information

The user information is needed in various different forms throughout the development process, from identifying opportunities to finalising the look of the application (Hyysalo 2009, 65.) The user information can be gathered from various sources, for example by analysing popular working methods and expectations of technology, using everyday common sense, conducting tests and interviews, observing users and adopting new ways of working (Oulasvirta 2011, 141). One method that comprises nearly all of the mentioned information sources is conducting a user research.

User research provides information about the users, such as their way of working, their desires, their frustrations and the problems they face. Even a limited understanding of users reduces the risk of producing unsustainable solutions. (Hyysalo 2009, 17.) Popular user research methods in system development include observations, surveys, interviews, tests, use case models and scenarios.

Observation means that a researcher monitors actions that are interesting for the research, without interfering the flow of actions. In participatory observation, the researcher not only passively follows what the examinees are doing but actively participates in their actions while making observations. (Saariluoma 2004, 38.)

Surveys may concern both the attitude of the user towards the new solution and the problems of the old system. With the aid of surveys, users can express their desires, future needs and also their disappointments. Interviews are another way to acquire reliable user information. Interview questions can be open or closed. If the topic is not well known, research-based open questions are appropriate. If exact answers are needed, closed-ended questions are convenient. (Saariluoma 2004, 42-43.)

User tests are valuable as having even one actual end-user as a tester can result in remarkable feedback to improve the service. The earlier the testing with an end-user is done, the greater is the benefit. The user tests can be done according to a predefined manual or without any guidance. When a manual is provided, the user performs the test by following the manual. These kinds of tests bring good feedback about the functionality and errors occurred during the assigned task. When no manual or instructions are given to the test user, the feedback can be more extensive and raise new issues that have not been thought. (Filenius 2015, 195.)

Use case models reflect how users interact with the system to achieve their goals. The model is used to define the different use cases of the users: goals of the users, their interactions with the system and the response of the system towards achieving the goals. (Eclipse 2012.) Scenarios are much like use case models. They define the stories and context of the users' actions and include the main pathways and alternative pathways the users use to accomplish their goals. (usability.gov 2017.)

2.3.2 Process flow

Once the general understanding of the users' needs, ways of working and processes have been identified, the outline of the application can be developed. The process continues with simplifying the processes and creating blueprints and wireframes to visually express how the solution would look like. Once the mutual agreement of the outline has been determined, the development of the application can be started. Many

of the smaller details are usually discovered and determined after the outline has been built and also tested. (Hyysalo 2009, 30.)

Limitations and opportunities can occur at any step of the development process, which needs to be taken into account at all stages of the design flow. For example, possible problems that may come up during implementation should already be considered in the previous concept design and technical design stages. (Hyysalo 2009, 56.) From time to time, some specifications are impossible to implement due to cost and technology constraints. Therefore, specifications should be kept open until their feasibility has been verified. (Hietikko 2008, 67.)

Many times a successful investment also requires more than one round of design, implementation, testing and trial run, known as iteration rounds (Hyysalo 2009, 56). Iteration rounds are needed for evaluating the usability and re-designing. The number of the rounds can be reduced by sufficiently addressing user needs, careful designing and careful consideration of solutions. (Jokela 2010, 62.)

2.3.3 Prototypes

Prototypes are preliminary versions of the application. The idea of prototypes is that each prototype created and tested helps to proceed further and leads closer to the final solution. In the early stages of development, it is not sensible to build heavy prototypes, because it takes time to build them and then the time to make the prototype can increase costs easily and unnecessarily. (Saariluoma 2004, 158.)

Specifications can change when prototypes created are being tested. New ideas and requirements may come up during testing of the prototype when the users can test the solution in practise. The purpose of making prototypes is not just to make one and implement it further, but to have and test other ideas. The best solution may be combined from the good sides of two prototypes. (Hietikko 2008, 67, 99.)

Solutions can also be sought by researching internally and externally. Solution models can be found on the internet and by enquiring from the users of similar solutions. (Hietikko 2008, 92.)

At some point in the process, the core solutions have made their form and no major innovations are being developed anymore. At this stage, the usability planning changes

to optimizing very little details. Then the most practical details and the best compromises for the big picture are found. (Saariluoma 2004, 160.)

2.3.4 Finalising a solution

When finalising a solution, it is good to consider motor coordination, cognitive capacity, emotions, communication and thinking (Saariluoma 2004, 160-161). Jacob Nielsen provides ten guidelines that are worth taking into account when defining the usability of an application:

1. Visibility of the system status: Does the system tell to the user in what stage it is?
2. Natural expression: Are the concepts and language used known and logical?
3. User in control: Are undo/redo, cancel and stop commands available to the user?
4. Coherent and standard: Is the look unified and standardised?
5. Preventing mistakes: Can the user make mistakes?
6. Identification rather than remembering: is the information needed found easily?
7. Flexibility and efficiency: Are there shortcuts?
8. Esthetical and minimal design: Are there any functions or elements the user does not need?
9. Error messages: Are the error messages in clear language? Do they contain a proposal of the solution or possibility to proceed?
10. Guidance and instructions: Are they logical and easy to find?
    (Jokela 2010, 57.)

Before launching the application, the guidelines above should be considered one by one. Firstly, the system status needs to be expressed logically. When a page is loading, there should be an hourglass or something else indicating that the process in ongoing. When clicking a button to go to a next page, the page should indicate into which page the user has come into. Secondly, the system needs to communicate in the user's language that must to be clear and straightforward. Thirdly, the control should be in the user's hands. Whatever choice the user makes, there should always be a clear exit and chance to cancel. Fourthly, the interface should be unified and standardised in a way that the content layout does not vary across pages or screens. Fifthly, the system should be designed in a way that the user cannot make mistakes. Sixthly, the user should be guided through the page without having to remember where some button exists. If instructions are needed, they should be easily accessible. Seventhly, shortcuts should be provided

when they help the user to process something faster. Eighthly, there should be no content visible that the user does not need. Ninthly, in case of an error message, the message should be expressed clearly in the user's language containing the reason why something went wrong and a solution how to solve the error. Tenthly, guidance and instructions should be easy to find, logical and concise. (Jokela 2010, 57.)

3 ECONOMY AND PROFITABILITY

After the usability problem has been identified and need has been recognised, it must be defined whether meeting the needs is beneficial in the perspective of profitability. It is known that SAP users use much time to perform their tasks and they are frustrated by the complexity of the system and waste of time. So the need is that the SAP users want to perform their tasks much faster, they want to enjoy using the system, have a better user experience and want to relocate the lost time to other tasks. The solution to these needs would be to develop a more systematic tool and a more user-friendly interface. But is it really profitable to build that solution?

The decision generally relies on three aspects: time, money and quality (Picture 1). Out of the three options, only two can be chosen. If a good quality solution with fast delivery is needed, it will not be cheap, and so on. Balancing between the three aspects is extremely difficult because everything is wanted. Good quality cannot be bargained because then the usability issues will not be fixed. Time relies on the availability of the project team members and users who want the solution as quickly as possible. The money aspect relies on how much money can be used. As good quality aspect cannot be touched, there is a bargain between the time and cost of the project. (The App Solutions 2016.)

Picture 1. Time, money and quality.

Enhancing the complex user interface starts from re-organising the user's desktop. A desktop is defined as the user's SAP screen: the interface where the user works on. The method applied for desktop clean-up is 5S that is part of the lean concept. 5S consists of five phases: Sort (clear and classify), Set in order (straighten and simplify), Shine (clean and check), Standardise and Sustain. (Lean Manufacturing Tools 2017.) It is a procedure used for organising work environments and standardising working methods. The objective of 5S is to improve productivity through identifying and eliminating the

losses and wastes from business processes and to remove the non-value-adding functions. (Lean Lion 2017.)

The 5S procedure can be applied as a project where the complex user interface is replaced with a more user-centric interface and then developed further as a new project when needed. Internal projects, such as enhancement and product development projects deployed inside an organisation, are intended to develop internal operations and produce functional and economic benefits. An internal project is successful when the cost savings achieved during a pre-determined period of time are greater than the costs of the project. (Eklund & Kekkonen 2014, 214, 219.)

The decision to start a new project, make an invention or produce a solution requires different analyses. No matter how useful in terms of employee motivation implementing the solution is, there is always the cost-benefit perspective. How much does the solution cost? Will it generate revenue? When will it pay itself back? What is the added-value? Then the calculations take place. (Eklund & Kekkonen 2014, 219.)

3.1 Development cost

Firstly, it is crucial to estimate how much the solution would cost. Typical costs include salary, travel, subcontracting, administrative and training costs, which can be either variable of fixed. Salary costs, such as planned working hours, time and incentive wages, travel expenses and subcontracting costs are variable costs. Administrative and management costs are fixed costs. To analyse the costs, the following questions should be considered: How many persons will participate in the project? How many hours of work is needed to implement the solution? What is the value of the hours spent in the project? (Eklund & Kekkonen 2014, 219-221.)

It is good to start with defining the people who are to be working on the project, the project team. In a software development project, a project manager is needed to control the project, a designer and a developer to implement the solution and testers and end-users to provide accurate information and testing. Then the number of hours each person involved is able to use on the project shall be defined. For example, a project manager can be capable of using 40 % of his/her working hours on the project, designer 15 %, developer 30 % and so on. The number of working hours needed and the duration of the project go hand in hand. If it is estimated that 200 hours of work is needed for the solution

implementation, then the duration of the project must be adjusted to the hours needed by the project team members.

The cost of the work consists of two factors: the amount of work done and the unit cost of the work. Determining them can be challenging as depending on the working situation the times may vary much. The total labour cost consists of gross wage and social security contributions. The salary paid to the employee can be 60 % of the total labour cost. The unit cost of the work is measured by the total labour cost. In some organisations, the unit cost of the work is a fixed total labour cost for a specific group of employees that is used in budgeting. (Neilimo & Uusi-Rauva 2009, 84-86.)

Some costs related to the project and development are difficult to trace. These costs include administrative costs and costs of using the software. Administrative costs are fixed costs that occur when such as time and advice is needed outside the project team. Costs of using software are generally license costs which are paid in an organisation or department level because the allocation of the costs can be difficult. There may be hundreds of people using the software from different teams which makes the allocation challenging. Therefore these costs are not part of the development cost estimation. (Bhimani et al. 2012, 35.)

The definition of development cost should not be sealed at the budget estimation phase. There are risks that cause budget overrun and often in software development projects original cost estimates are exceeded. There are two reasons for budget overrun of which the first is that system specifications can be difficult and challenging to implement. This can be due to that documented requirements are indefinite and defined clearer during the process. Also, naturally new thoughts and requirements are added and older ones are wiped away. As the project lives all the time, the final outcome may not remind much the original design. (Filenius 2015, 193.)

The other reason is related to buying or investment decisions. The cost estimate is typically reduced so that decisions can be made. The estimate can be too optimistic and attractive that possible risks for exceeding the budget are not considered. In these cases the buyer usually knows that the amount to be invested is not enough to cover the cost of the project. (Filenius 2015, 193.)

It is important to make the cost estimate as accurate as possible by taking all risks and incidents into account. The further the development project has proceeded, the larger

the costs for changes are and suspension of the project will cause major losses. (Hietikko 2008, 48.)

3.2 Profitability

Investments, such as software, are assets acquired by an organisation for a long period of time. Investments are directed to the future and intended to increase sales or produce cost savings. In order that the new user interface becomes profitable, the costs of implementing the interface need to be smaller than the benefit. To estimate the benefit, there are several methods to calculate the profitability of the investment: the payback time, accounting rate of return, net present value and internal rate of return. Payback time and accounting rate of return are quick and simple to calculate and they are intended to analysing investments that have a short lifetime. Payback time calculates the time how fast the investment pays itself back. The accounting rate of return measures the average annual rate of return of the investment. These two methods, payback time and accounting rate of return, do not take into account the time value of money. They only consider the value of the investment till the invested cash has been retrieved. (Braun et al. 2010, 670.)

Payback time is the time the investment pays itself back and the time during which the expected net incomes of the investment cover the cost of the investment. The payback time is calculated by dividing the cost of the investment by the expected net income (Equation 1). If the payback time is shorter than the economic lifetime, the investment is profitable. The economic life of an investment is the period of time during which the investment is expected to be in use and generate income to the business. If the payback time is longer than the expected economic life, then the investment generates loss. The payback method is applicable especially for profitability calculations of short-term investments, as it does not consider the time value of money. (Eklund & Kekkonen 2014, 37, 138.)

$$Payback\ time = \frac{\text{Initial investment}}{\text{Expected net income}}$$

Equation 1. Payback time (Eklund & Kekkonen 2014, 37).

The accounting rate of return (ARR) measures the average annual rate of return over the investment's economic life. Instead of considering the net income an asset generates, the ARR focuses on the operating income. It is calculated by dividing the average annual operating income from the asset by the initial investment (Equation 2). The operating income is formed by subtracting annual depreciation expense from average annual net income. Annual depreciation expense is computed by dividing the subtraction of initial cost of the asset and residual value by the asset's economic life in years. (Braun et al. 2010, 675.)

$$Accounting\ rate\ of\ return = \frac{\text{Average annual net income} - \text{Annual depreciation expense}}{\text{Initial investment}}$$

Equation 2. Accounting rate of return (Braun et al. 2010, 675).

Depreciation is the reduction in the cost of a fixed asset. Its purpose is to allocate the cost of the investment to all the time periods during which the asset is expected to generate income. Depreciation measures the periodically detrition of an asset. The factors that affect the depreciation are the depreciation period, the value of the asset, allocation of depreciation and residual value. (Neilimo & Uusi-Rauva 2009, 97-99.) The residual value is the value of an asset at the end of its useful life – the value received when the asset is sold (Investopedia 2017).

The accounting rate of return is based on accrual accounting. It uses the asset's operating income instead of the asset's net incomes. Typically, companies set a minimum accounting rate of return that is required to be exceeded in order to invest in the asset. It is used by companies to measure the asset's profitability over its economic life. The higher the ARR, the more desirable the asset is. The method is convenient to use in analysing an investment's cost-effectiveness, but, it does not consider the time value of money. (Braun et al. 2010, 676-677.)

The inflation and the uncertainty associated with the future affect on the time value of money and the expected income of the investment. The time value of money means that the money available today is worth more now than the same amount in the future. (Braun et al. 2010, 679.) Due to the inflation and uncertainty, a discount rate is used in the investment calculations to convert the value of the future net incomes to a value equivalent to the current value of money. This process is called as discounting to the present value. (Eklund & Kekkonen 2014, 37, 133.)

The net present value (NPV) is a discounted cash flow model and it measures the difference between the present value of the investment's net income and the investment's cost. The NPV is calculated by discounting the annual net income to the present value (PV) and then subtracting the value by the cost of the investment (Equation 3). To compute the PV, Annuity of 1€ is used and i as the percentage PV factor and n as the economic life in years must be known. The result of NPV must be positive in order that the investment is profitable based on the calculation. (Braun et al. 2010, 686-687.)

$$\begin{aligned} &Net\ present\ value \\ &\quad = \text{Annual net income} * (Annuity\ PV\ factor\ for\ i = \%, n = years) \\ &\quad - Initial\ investment \end{aligned}$$

Equation 3. Net present value (Braun et al. 2010, 686).

The internal rate of return (IRR) is the rate of return when the net present value is equal to zero, NPV = 0. The IRR measures the exact rate of return that an investment composes (Equation 4). High internal rate of return means a more desirable investment. The IRR is calculated with the equation: cost of the investment equals present value of the investment's net income. (Braun et al. 2010, 691.)

$$Internal\ rate\ of\ return = \frac{\text{Investment's cost}}{\text{Net income}} - Annuity\ PV\ factor\ (i = ?, n = years)$$

Equation 4. Internal rate of return (Braun et al. 2010, 691).

The reverse event for discounting is interest rate calculation. The interest rate defines how much more valuable the amount of money is today than after a certain amount of time. It is the imputed rate of interest that represents the minimum return expected from the investment. The interest is calculated by discounting the future amount of money to the present day by using an agreed interest rate. In Equation 5, the reverse event of interest rate and discount rate is shown: i is the interest rate/discount rate and n is the number of years. (Neilimo & Uusi-Rauva 2009, 216.)

$$Interest = (1+i)^n \; vs. Discounting = \frac{1}{(1+i)^n}$$

Equation 5. Interest rate versus discount rate (Neilimo & Uusi-Rauva 2009, 216).

To sum up, when evaluating the cost of the investment, the following aspects are to be measured and identified: the acquisition cost of the investment, the useful/economic life of the investment, the residual value, the estimated future net payments and net income and the rate of interest. The acquisition cost of the investment must be smaller than the net residual value plus the difference of the estimated future net incomes and net payments during the investment's economic life in order that the investment becomes profitable. (Jormakka et al. 2015, 230.)

4 CASE WÄRTSILÄ

At Wärtsilä, it has been discovered that some employees find it difficult and time-consuming to use SAP as the enterprise resource planning tool. To solve the usability issues and enhance the user satisfaction, Wärtsilä has considered taking in use SAP add-on software. The add-on examined is Screen Personas.

4.1 Background

Wärtsilä, founded in 1834, is a manufacturer and service provider of complete lifecycle solutions for global marine and energy markets. In 2016, Wärtsilä had approximately 18 000 employees in over 200 locations in 70 countries across the world. Wärtsilä's net sales in 2016 were 4.8 billion euros. The organization consists of three businesses: Marine Solutions, Energy Solutions and Services. (Wärtsilä 2017.)

SAP is an enterprise resource planning system that combines different business functions in a real-time environment. SAP consists of applications, also known as modules, such as sales and distribution, inventory management, production planning, finance and controlling, customer relationship management and human resources. The modules can be taken into use one by one according to the needs of the business. The purpose of the ERP systems is to help organisations to manage their businesses, streamline the business' operations and unify procedures and policies. (Jormakka et al. 2015, 253-254.) The systems can be used to improve strategic and operational planning, speed up decision-making, identify risks and forward real-time information, to mention a few (Tutorialspoint 2017).

SAP ERP has been in use at Wärtsilä since 2004. Today economy, logistics and customer projects are managed through the system. During the years, Wärtsilä has grown and acquired companies to join Wärtsilä. The acquired companies have been using many other different systems before coming to Wärtsilä. To ensure consistency, they are taken into the SAP environment. New systems along with new processes and new ways of working can be challenging to adopt. One potential way to ease to adaptation of SAP is to build new and more user-friendly interfaces for the newcomers that are developed for their specific use.

The business unit for whom the case is implemented consists of warranty handlers who register and handle warranty claims. The business unit has come to SAP via a joint venture in 2012. The warranty handlers find it difficult, frustrating, and time-consuming to handle the customer warranty claims in SAP. Therefore, they were a good target group for whom to build the new user interface with Screen Personas.

4.2 What is Screen Personas?

Screen Personas is a browser-based SAP platform used for creating simplified and personalised user interfaces from SAP GUI (Graphical User Interface) screens. Screen Personas provides drag-and-drop and hide features that can be used to move objects and hide unnecessary objects from the SAP screens. Also, scripting is enabled. (SAP 2017.) Personalisation and customisation of SAP screens for specific transaction codes or business processes can decrease the overall time a user spends on performing the process, cut down data entry errors and reduce the number of inputs and clicks needed for task execution (SAP 2016).

There are multiple benefits and advantages that can be gained with Screen Personas. Fewer screens, fewer clicks, fewer data inputs, standardised user experience and reduced training time and cost are benefits that Screen Personas can deliver along with improved usability, process efficiency and user productivity. (SAP 2017.) Other unmeasurable benefits that can be achieved with Screen Personas are increased user motivation and satisfaction and better adaptation of SAP as the ERP system.

The three main functionalities of Screen Personas are scripting, modifications of screens and theming. Screen Personas supports scripting with JavaScript. The functionality allows recording keystrokes, doing computations, pre-executing steps and skipping screens. Complicated scripts have a greater impact on performance but also a longer time of rendering the desired action. (SAP 2016.)

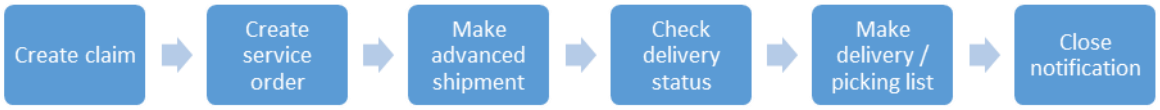
Screens can be modified and tabs can be merged in Screen Personas. The fields and buttons can be moved from one place to another, which means that information from several tabs can be joined in one screen or a fewer number of tabs. Merging information enables the user to see the information needed faster and to access it with fewer clicks. For example, instead of the user needing to enter or retrieve data from eight different

tabs, the data could be accessed only from one to three tabs. Also, changing of field descriptions, colours and fonts is enabled. (SAP 2016.)

Themes are used to change the appearance of a transaction or multiple transactions. With the theming functionality screens within a specific business process can be made consistent so that each screen has the same background, same colours for controls and objects, same border style etc. By theming, the user can identify more easily the specific process among other processes.

4.3 Identifying the need and context of use

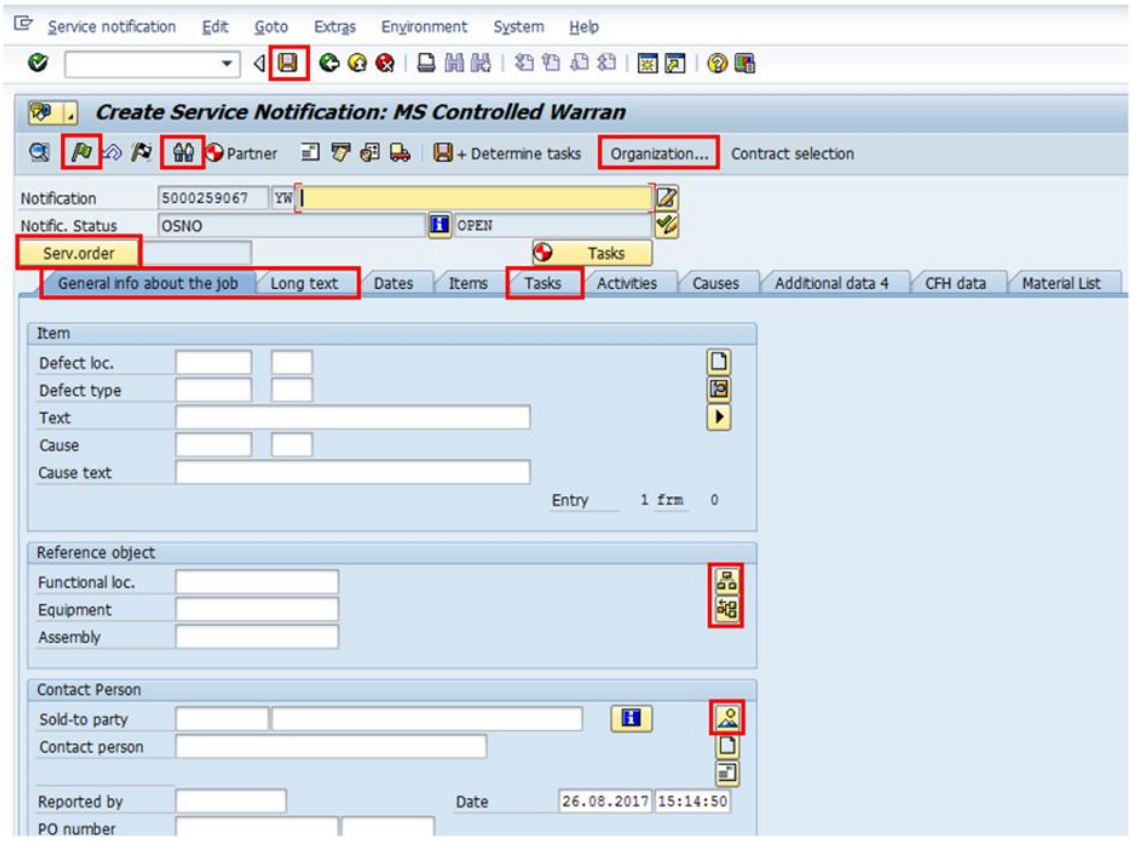
The process for handling warranty claims generally consists of six main steps (Picture 2). The first and last steps, creating and closing claim, are valid to all warranty notifications, but the four middle steps, creating service order, making advanced shipment, making delivery and picking list and checking delivery status, may not appear in all warranty claims depending on the nature of the case.



Picture 2. Warranty handling process.

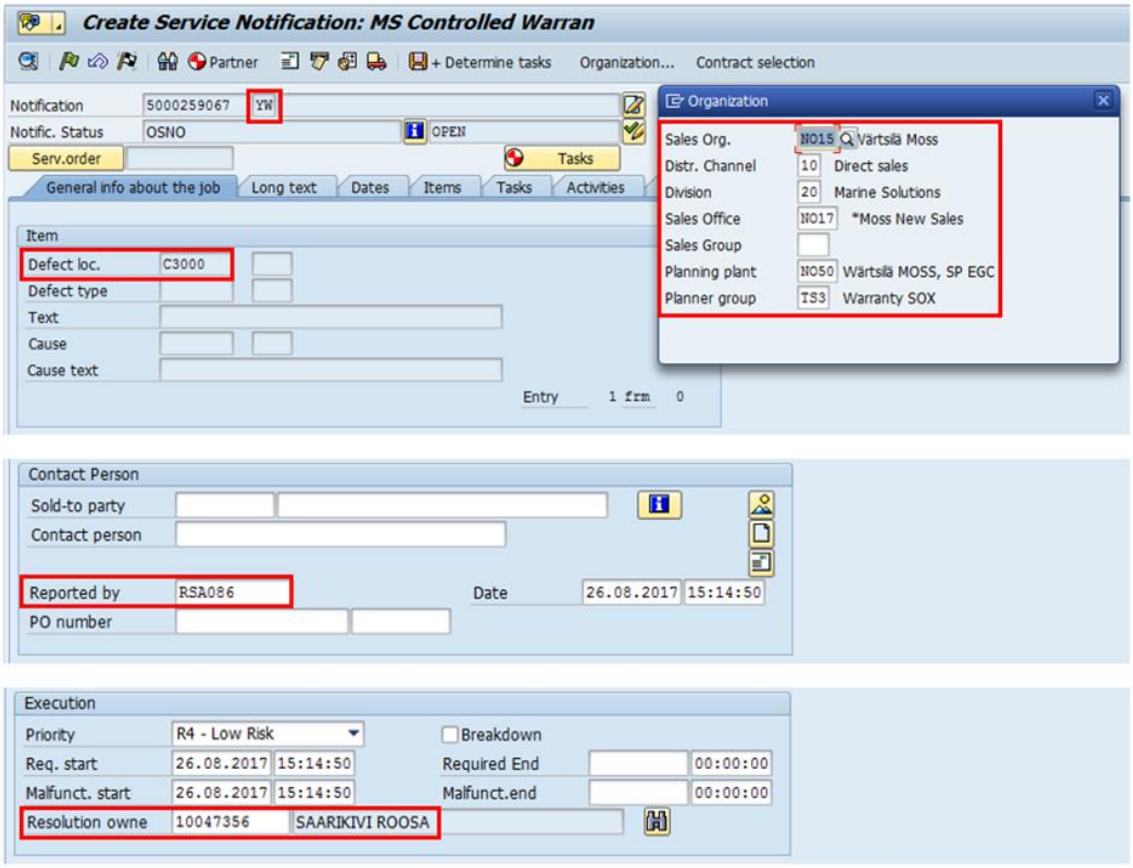
The process starts with a customer claim. During warranty period, a component has been damaged and it needs to be replaced free of charge on behalf of Wärtsilä. The customer sends a notification which is received by a warranty handler at Wärtsilä. The warranty handler registers the claim into SAP (Step 1). As a new component needs to be sent to the customer, a service order will be created by the warranty handler (Step 2). The list of components is added to the service order, which then reads from the system if the component exists in the warehouse. If there is no component available in the warehouse, advanced shipment needs to be made (Step 3), which is generally a purchase requisition. To know when the ordered part has arrived to the warehouse, the delivery status needs to be monitored (Step 4). Once the ordered part has arrived, it will be sent to the customer by informing the warehouse with delivery and picking list (Step 5). Finally after the customer has received the new component, the claim can be closed (Step 6).

Totally, the warranty handling process requires entering seven different transaction codes and at least 20 different tabs and screens. The transaction codes consist of creating claim, changing claim, creating service order, changing service order, creating sales order, checking the delivery status and changing sales order. Within these transactions the user needs to go through several tabs and screens such as to input the basic information, determine tasks, enter stakeholders and enter components. In addition to the tabs and screens the warranty handlers use, there are even more tabs and information visible that they do not need, so the amount of information the warranty handlers see is massive. For example, when registering a claim in the system, the warranty handlers have ten tabs visible on the screen of which they only need three. In addition, there are a few dozen buttons available to be clicked, but of which they only use around ten. In Picture 3, the tabs and buttons the warranty handlers use when registering a claim are circled in red. As noted, there are many other tabs and buttons visible that are not needed when registering a claim. All transactions, tabs, screens and buttons used by the warranty handlers contain numerous fields of information of which most are blank and unnecessary for the warranty process. Only around 10 % of the fields are relevant to the process. The unnecessary fields visible can cause data entry errors and produce inconsistent analytical data.



Picture 3. Tabs and buttons used when registering a claim in SAP.

Another frustrating issue in addition to the massive information stream towards the warranty handlers is that every time they go through the process they enter values that are always same for all warranty cases, such as notification type, organisation data, main work center, payment terms to mention a few. In Picture 4, the fields circled in red are values that are always the same and need to be input whenever registering a claim. To input these 10 values, it requires almost 30 clicks, data inputs and enters to do it. When inputting the values manually, there is a risk for entering false data which, if not noticed and fixed, can cause unreliable analytical data.



Picture 4. Fields that are same for all warranty cases when registering a claim.

Also, some pieces of data that should be the same across transactions are not automatically copied from a transaction to another. The warranty handlers need to enter this data themselves which is time-consuming and risky for data-entry errors. In total, entering these values that are always same and could be automatically defaulted take over hundred clicks, enters and data inputs throughout the warranty handling process.

Handling of one warranty claim takes average 8 hours (one day) of the working time of a warranty handler. It is an average value as some warranty cases can take a few hours and some several days for their handling. A claim can also be open in SAP for over a month for example when it is waited that a ship is docked the next time. Half of the average 8 hours the warranty handlers spend on registering and processing the claim in SAP.

After having defined the problem and the context of use, there is an opportunity to make the process simpler and faster by combining necessary information in tabs and screens

to a fewer amount of screens, hiding unnecessary information, automating steps and defaulting values.

4.4 Plan, budget and cost

The objective of the project is to simplify the warranty management process in SAP with a more personalised and customised user interface. The scope of the project is constricted to two Wärtsilä plants with a few warranty handlers.

The estimated duration of the project is eight weeks and the number of hours planned is 114 hours. Eighty hours is estimated for planning and developing the user interface, 30 hours for two end-users to provide need specifications and to do testing and the final 4 hours for support from IM (Information Management) organisation.

The estimated interest-free cost of the project is 4 860 euros (40 €/h for planning and development, 50 €/h for end-users and 40 €/h for IM consult). The hourly rates used are total salaries per hour including such as gross wage, annual pays and sick pays. The rates are used in budgeting defined by Wärtsilä. The possible risks to exceed the estimated cost are that the development of the interface takes more time than expected and the project prolonging. It has been estimated that the user interface is ready after three iteration rounds, each lasts one week. If more iteration rounds are needed, the developer uses more time to the work than expected and also the end-users need more time for the testing. If the number of the iteration rounds are doubled, the estimated cost can rise by 3 000 euros.

4.5 Benefit estimates

The sought benefits of applying the solution of a new user interface include not only measurable but also non-measurable benefits. The measurable benefits looked for are saved time, increased efficiency, decreased amount of data entry errors and improved data quality. Saved time and efficiency are measured by comparing the amount of hours the warranty handlers spend in SAP now and in Screen Personas afterwards. The change in the amount of data entry errors can be measured by analysing the probability of making mistakes when reducing the number of fields and defaulting data. The quality of the data can be measured by the consistency of the data through defaulting values.

The non-measurable benefits sought by making the processing of warranty claims easier and more pleasant include reduced frustration, improved motivation and better user experience. The evaluation of these benefits are outside the scope but important to mention.

The project is not intended to increase sales but to save on costs, cut down errors, improve data quality and make the workflow easier. When warranty handlers spend less time on non-value-adding tasks, they will have more time on other tasks and on giving better customer experience. In addition, the data quality is improved as the information needed is more easily accessible and the risk for data errors is decreased.

As the benefits are mostly measurable, they are estimated as hours and then converted to money. In this study, example values are used in the calculations. The values do not represent the actual values, but are invented. By replacing the values with actual values, precise calculations can be made.

Before estimating the profitability, the present values of how many claims are registered in a specific period of time, how much time it takes to process one claim in SAP, how many data errors occur per each claim and how much time it takes to solve these errors must be defined. In this case, an invented value of 400 is used as the number of claims registered and a year is used as the period of time. As already defined, it takes 8 hours on average to handle one claim and half of that time (4 hours) is spent in SAP to register and process the claim. Based on the invented values of 400 claims and one year, the warranty handlers spend 1 600 hours in SAP in a year to process claims. Also, in this example case, an invented value of five is used as the number of errors occurring during each claim and 15 minutes as the time it takes to solve the errors in order to proceed further. The example time the warranty handlers use to solve the errors in a year is then 100 hours.

Once the new user interface is ready to be taken into use, it is estimated that the time the warranty handlers spend in SAP Screen Personas to register and process one claim on average is reduced to 3 hours and 30 minutes. So the benefit is 30 minutes per claim. The estimation is evaluated by the time expected to be saved by a warranty handler when some values are defaulted, some process steps are automated and the risk for making errors is reduced. For example, the notification type and organisation as defined earlier in Section 4.2 are to be automatically defaulted which is estimated to reduce the time on inputting values, some unnecessary pop-ups and screens are to skipped by

scripting that would decrease the time to proceed to the next step and the risk for making errors is reduced by decreasing the number of input fields in addition to defaulting values so that there will be less room for data entry errors. The influence of improved workflow and better data quality are not measured here.

When using the example values of 400 as the number of claims and one year as the period of time, the example total benefit is estimated to be 200 hours (400 * 0,5 h) of saved time in a year. When the hours estimated to be saved are converted to money (1 h = 50 €), the cost saving is 10 000 euros in a year in this example case. The new user interface is expected to be used for 2 years and after the two years it will have no residual value. The annual straight-line depreciation is 2 430 € (4 860 € / 2) and the annual depreciation percentage is 50 %.

To estimate the profitability of the project, the payback time, accounting rate of return, net present value and internal rate of return are calculated. The payback time of the project is calculated by dividing the cost of the investment (4 860 €) by the example expected annual net income (10 000 €). The payback time is then 0,49 years, so the investment would pay itself back in less than 6 months. If the cost estimate of the project is exceeded by 3 000 € because of more iteration rounds needed, the cost of the investment would then be 7 860 €. The payback time is then 0,79 years, so if more iteration rounds are needed, then the project would pay itself back in less than 9,5 months.

The accounting rate of return measures the average annual rate of return over the investment's economic life. The annual depreciation of the investment is 2 430 € with an economic life of 2 years, the cost is 4 860 € and it will have no residual value. The ARR is calculated by dividing the average annual operating income by the cost of the investment. The ARR is then 156 %. If the estimated cost is exceeded by 3 000 €, the annual depreciation is then 3 930 € (7 860 € / 2) and the ARR is 77 %.

The two calculations above give an estimation about how fast the investment will pay itself back. However, they do not take into account the time value of money and what is the revenue after the investment has paid itself back. Therefore, the net present value and internal rate of return are also calculated.

The net present value measures the difference between the present value of the investment's net income and its cost. The applied discount rate used in the calculations is 9,0 %. The percentage is an average discount rate used in Wärtsilä's quarterly reports.

With the cost of the investment of 4 860 €, example expected annual net income of 10 000 €, economic life of 2 years and discount rate of 0,09, the net present value of the investment is 12 731 €. If the estimated cost is exceeded by 3 000 €, then the net present value is 9 731 €.

The internal rate of return is the exact rate of return that the investment composes, when net present value is equal to zero. As the estimated cost of the investment is 4 860 €, example expected annual net income is 10 000 € and economic life is 2 years, the internal rate of return is 180 %. If the estimated cost is exceeded by 3 000 €, then the internal rate or return is 93 %.

All example profitability estimates are gathered in Table 1 to visually represent and compare the estimates. 10 000 € is used as the net income, 2 years as the economic life, 50 % as the annual depreciation percentage, 0 € as the residual value and 9,0 % as the discount rate.

Table 1. Profitability estimates.

COST	PAYBACK	ARR	NPV	IRR
4 860 €	< 6 months	156 %	12 731 €	180 %
7 860 €	< 9,5 months	77 %	9 731 €	93 %

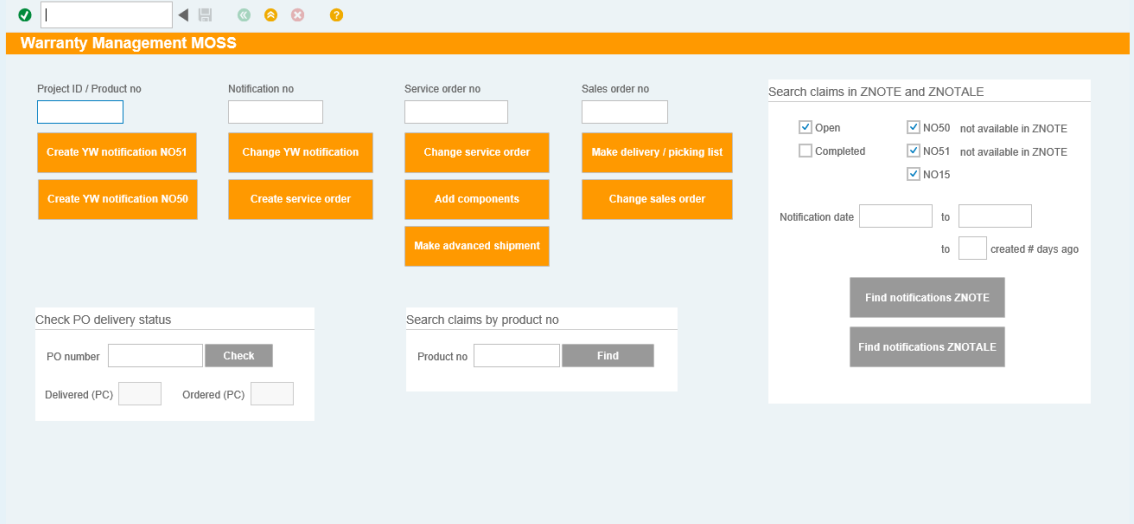
All example calculations show either a positive euro value or a high percentage, which means that the investment is profitable based on the estimates. As the payback time is less than a year and the economic life of the investment is two years, the investment pays itself back during its useful life, so it is worth investing. Also, the net present value is highly over zero, so the investment should also generate profit after it has been paid back. The accounting rate of return and internal rate of return show a high percentage which indicate that there is less risk that the investment becomes unprofitable.

4.6 Development

Once the decision to kick off the implementation of the solution has been made, the process walkthrough begins. As some of the need specifications and contexts of use are

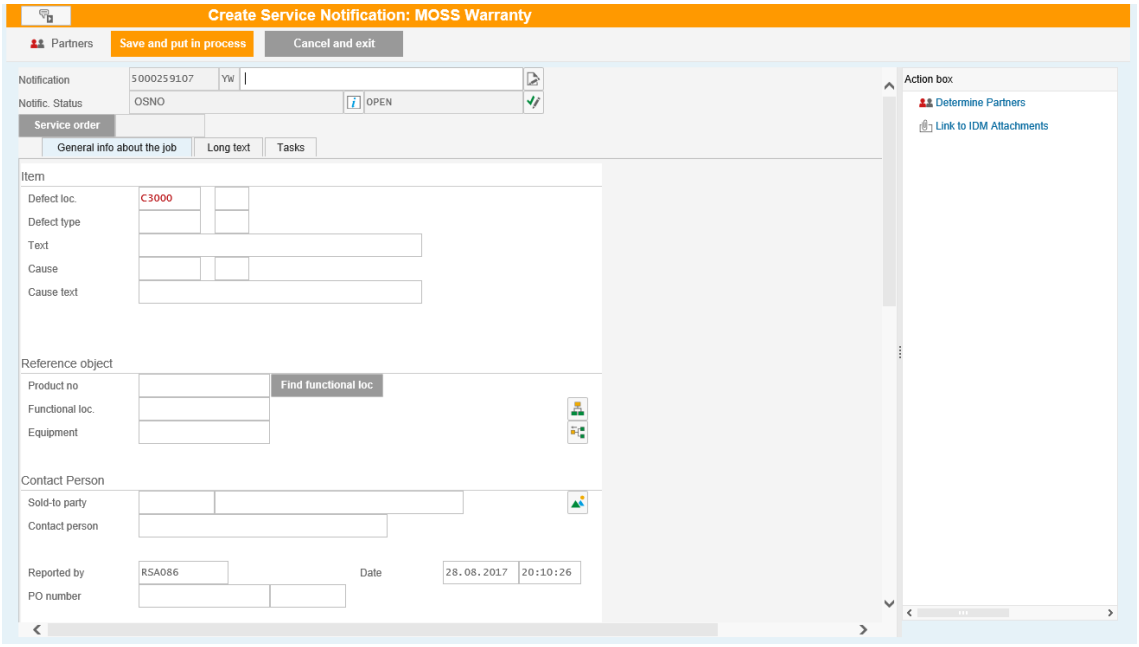
already familiar, the process how warranty handlers currently perform the tasks is to be studied in detail: Which buttons do they use? What pieces of information do they need? What pieces of information do they not need? The process walkthrough requires time and collaboration from both parties, the developer and the end-user, to fully understand what is required.

When the developer is fully aware of the process description and the requirements of the users, the layout design is being formed. The layout design gives the outline for the interface. As already defined earlier, the warranty handling process consists of six steps. Logically thinking these six steps could also form some of the main buttons. To make the process overview simpler, there could be a common desktop filled with necessary buttons to go through the process. Based on the process walkthrough and the requirements gathered from the users, the desktop was built in a way that it comprises all necessary fields and buttons to perform the warranty handling tasks, as seen in Picture 5.



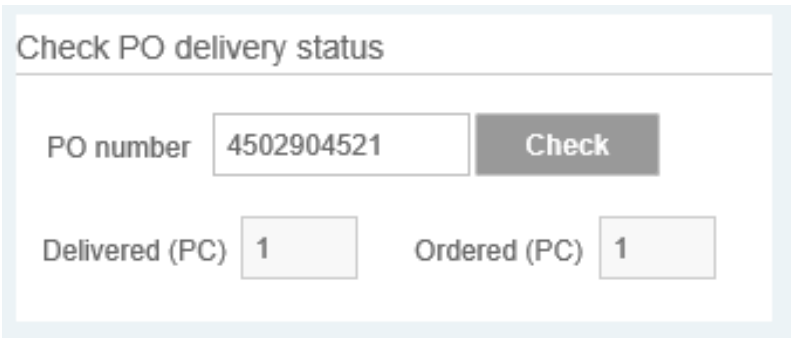
Picture 5. Final view of the desktop in the new user interface.

After the layout design, also the transactions and screens were modified and made simpler. As there was too much irrelevant information visible on the screens, the screens were re-created and only the necessary information was left visible in a fewer amount of tabs. For example, the transaction where warranty claims are registered consisted of ten tabs and multiple unnecessary fields and buttons. In the new user interface, the unnecessary tabs, fields and buttons were hidden and only the necessary ones were left as seen in Picture 6.



Picture 6. View of the claim registering screen in the new user interface.

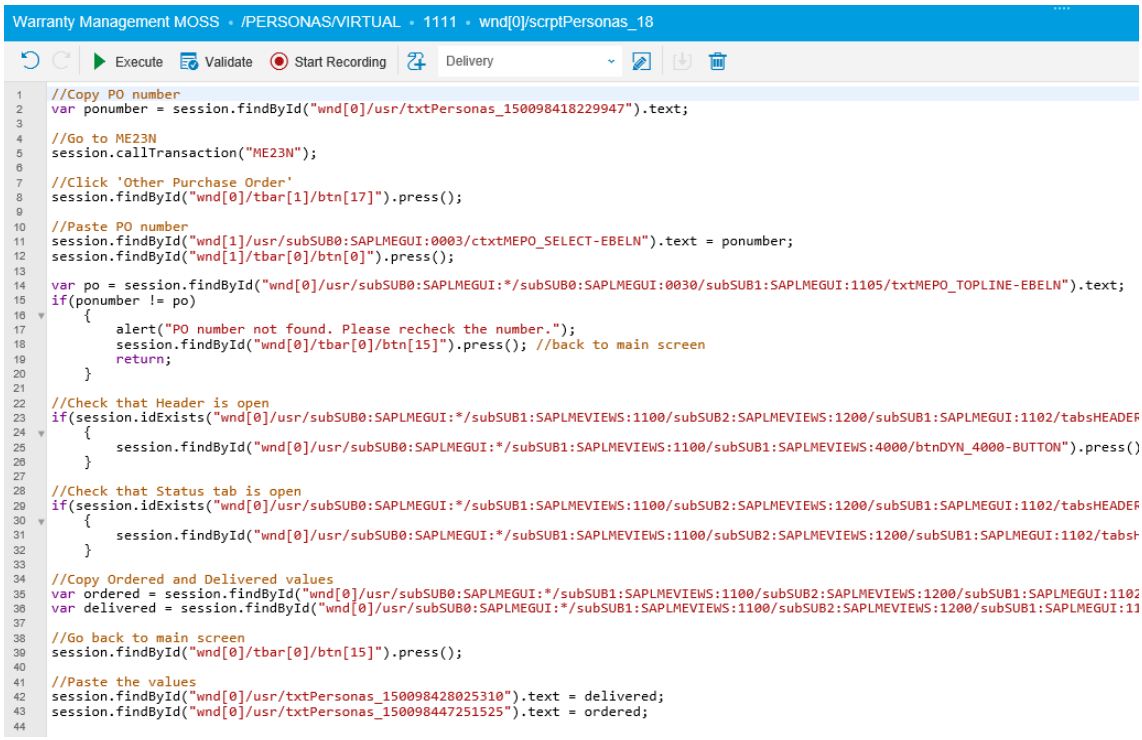
New buttons were added to the screens to automate the warranty handling process. In these script buttons, the developer was able to add code in order to automatically default values, skip screens and pop-ups, automate steps and prevent data-entry errors. One simple script was written to ease the monitoring of delivery status. To perform the task in SAP, the user would first need to go to the specific transaction, enter the purchase order number and navigate to the status information. In total, it takes viewing three different screens and several clicks and enters to check the information. With Screen Personas it has been able to automate the steps in a way that the user only needs to know the purchase order number, enter it to a text field and click one button as shown in Picture 7.



Picture 7. Delivery status check automated in Screen Personas user interface.

4.7 Challenges in building user interface in Screen Personas

Screen Personas supports JavaScript as the scripting language. The script functionality allows recording keystrokes, doing computations, pre-executing several steps and skipping screens. (SAP 2016.) As seen in Picture 8, the delivery status check was automated by scripting which only needed a few dozen lines of code.



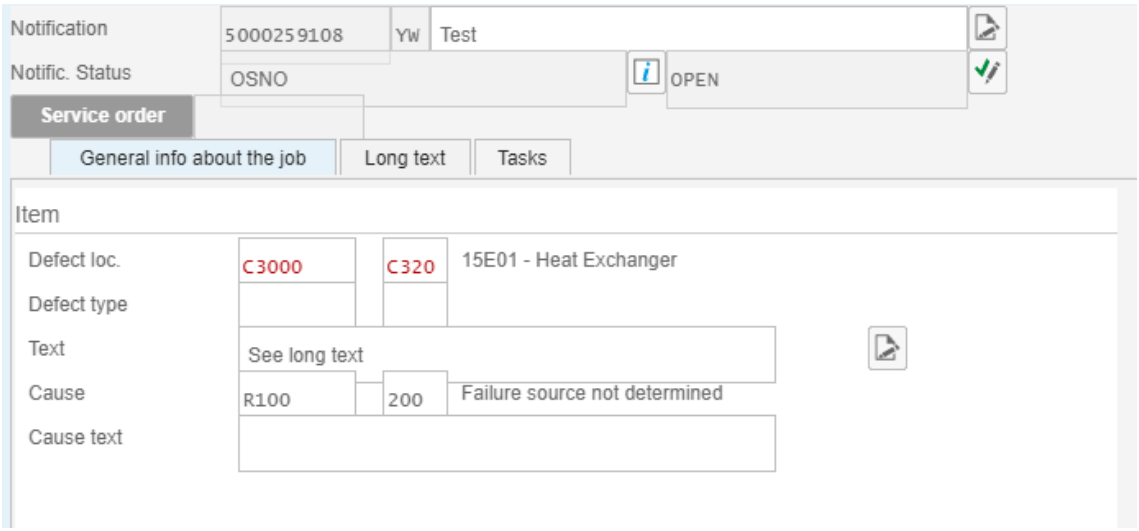
Picture 8. Script generated in Screen Personas to automate steps.

When building the new user interface in Screen Personas, over 20 different scripts were created to optimise the processing of warranty claims. Most of the scripts were large and went through several screens and transactions while executing commands and copying and pasting information. What was noticed while executing the scripts, there was not a large time difference in rendering an action based on the size of the code. The greatest difference regardless of the size of the code in rendering was the fact how deep the code went in SAP. For example, in one of the process steps the user can enter two transactions inside one transaction: First the user enters ‘Change claim’ transaction. There the user clicks ‘Change service order’ and finally ‘Advanced shipment’ transaction without exiting ‘Change claim’ and ‘Change service order’ transactions. The time it takes the script to execute is really long even the script is very simple, only ten rows.

No matter if there is no script behind when entering the ‘Advanced shipment’ through the two other transactions, the rendering of the screen is nevertheless slow. In SAP the response time can be less than two seconds but in Screen Personas it may be from 5 to 15 seconds. Inputting information and changing tabs is also slow in that transaction. The outcome is that the further navigating in SAP, the slower and slower the response time gets.

Another factor that has a big influence on the response time is the modifications made to the screens and transactions. When there are no changes made, the response time in Screen Personas is nearly as fast as in SAP. If making several modifications, such as combining tabs and changing places of fields, the response time slows down with the changes. So the more the screens are being modified, the slower they become.

Screen Personas has several other limitations not mentioned in the SAP websites that came across during developing the user interface. Firstly Screen Personas is not compatible with all most popular web browsers. Google Chrome has become a very popular web browser at Wärtsilä. In many business related tasks, it is much faster than for example, Internet Explorer. However, Screen Personas is not entirely compatible with Google Chrome. As seen in Picture 9, some fields strangely overlap other fields. In Internet Explorer the fields adjust to the layout perfectly, which means that the user cannot select between browsers but has to settle for the slower Internet Explorer.



Picture 9. Field adjustment in Google Chrome browser.

In addition, organisation transaction codes can produce problems in Screen Personas. These transaction codes are created for processing a specific business process inside

an organisation apart from the standard SAP transactions. When organisation transactions are entered in Screen Personas undefined error messages can pop up even though there has been no modifications made in Screen Personas in these transactions. Neither the developer nor the user knows why they come from. Also, the error messages can be difficult to reproduce in order to solve the problem.

The path of a new user interface within the SAP system is from development to test and then to production. The interface is first developed in the development client and once ready to be tested it is transported to the test client. Sometimes the transport can cause errors in the scripts. The script that has worked correctly in the development client has now errors in the other client. The error can be that some object has a different ID compared to the other client. Because of it, every single script shall be re-tested and possibly fixed, which takes extra time from the developer.

4.8 Feasibility

Feasibility is measured by defining the can dos and cannot dos of Screen Personas. In Table 2, the pros and cons of developing a new user interface in Screen Personas are identified.

Table 2. Pros and cons of Screen Personas.

PROS	CONS
• Unnecessary site content can be hidden • New fields and buttons can be added to enhance the task processing • Fast script execution • Easy to write scripts: recording of key-strokes is enabled • Only basic knowledge of JavaScript needed • Easy to move information and create new content • Can be applied to both small and big user groups	• Modifying the screen is slow • If there is an error in the script, the pop-up error message shows as a “Backend error occurred” and does not provide any solution to solve the problem • The further going within transactions, the slower the response time gets • Creation of new tables is not enabled • Large tables are not requested to be moved from a tab to another as it slows down the performance significantly

As noted, Screen Personas has weaknesses but also multiple strengths. One of the most value-adding functionalities is scripting. With scripting, steps can be automated and values can be defaulted. It not only increases the efficiency but also ensures a better data quality. When the data is consistent across claims registered, the data can be used to provide accurate and reliable data for analytical purposes.

However, Screen Personas is not entirely applicable for implementing one-screen-processes, which would have been a very beneficial advantage. As moving tables reduces the response time significantly, tables needed in the processing must be left at their tabs and so there is more than one tab and one screen visible to the user.

4.9 Final product and realised benefits

One way to evaluate how the usability was taken into account at the final product is to consider the ten guidelines of Jacob Nielsen. According to Nielsen, the system should

tell to the user at what stage it is. The solution developed follows the guideline: when a page is loading there is an element indicating the processing. Also, each page and screen has a corresponding title indicating at what step of the process the user is. Secondly, the language used in the system is English, which is the common working language at Wärtsilä, and the information is expressed in a logical order: first registering a claim, then changing the claim etc. Thirdly, the user has the possibility to go back, exit and cancel at any stage of the process. Fourthly, the look is standardised by Fiori theming, which is commonly used at Wärtsilä. Fifthly, most of the mistakes a user can make are prevented by scripting and by providing short tips to the user. Some mistakes have not been able to be prevented due to the constraints of SAP. Sixthly, the information needed is found easily and the information the user does not need is hidden. Seventhly, there are multiple shortcuts in the new interface. Steps that do not add value to the claim processing have been skipped which also makes the processing of the claim faster. Eighthly, all unnecessary fields, buttons and tabs have been hidden so that the content that is only relevant for claim processing is visible. Ninthly, error messages generated by scripting are messaged in a clear language containing the reason why the error occurred. Tenthly, the guidance for using the new interface is embedded in the system. When needed more precise and larger instructions the manual is provided in internal document management system.

When creating the project schedule, it was estimated that the project takes 8 weeks. The first two weeks were budgeted to process walkthrough and requirements collection, the third week to layout design, the following 3 weeks to the development of the user interface and the last two weeks to the creation of user manual and go-live. The budgeted hours for the team were 114 hours in total consisting of 80 hours for planning and development, 30 hours for the specifications and testing and 4 hours for support.

The realised hours exceeded the estimates as taken into account as the most possible risk. The realised hours exceeded by 60 hours. Fifty hours of more time were used to the development of the user interface and 10 hours more to the testing. The cost of the project then exceeded by 2 900 €, and the realised budgeted cost was 7 760 €.

The reason for exceeding the cost estimates was due to the slowness of Screen Personas, unexpected errors and issues, fixes needed after transporting the interface across SAP clients and new user requirements. The slowness of Screen Personas mostly occurred during modifying the user interface. When going to the ‘edit mode’, the response time of Screen Personas slowed down and it took much more time than

expected to hide irrelevant fields and tabs and to move the fields and information from one place to another. Unexpected errors and issues included disappeared and re-settled fields that took a considerable amount of time to be fixed. When transporting the interface across clients, from development to test and then to production client, some added buttons disappeared and some scripts did not work anymore. Everything needed to be double-checked and fixed. The new user requirements led to least extra hours. The requirements were small and mostly contained script modifications and button changes which were easy to implement.

Due to the slowness of Screen Personas, the estimated time savings were not met. It was estimated that executing one warranty claim in Screen Personas would take 3 hour 30 minutes instead of the 4 hours in SAP. Instead of 3 hours and 30 minutes, it takes 3 hours and 40 minutes to perform the process with the new user interface in Screen Personas. The benefit is then 20 minutes per claim registered. When using the example values of 400 as claims registered and one year as the period of time, the total benefit is then 133 hours of saved time in a year. When the hours saved are converted to money (1 h = 50 €), the cost saving is 6 650 euros in a year in this example. The expected economic life is the same, 2 years. As the cost of the investment increased and the saved time decreased, the profitability shall be re-calculated.

When using the example values of 400 as the number of claims and one year as the period of time, the new payback time for the investment with the cost of 7 760 € and annual net income of 6 650 € is 1,17 years. The investment would pay itself back in 1 year and 2 months. The new accounting rate of return with an annual depreciation of 3 880 € is 36 %. The new net present value when using the discount rate of 9,0 % is 3 938 €. The new internal rate of return is 45 %. The values are represented in Table 3.

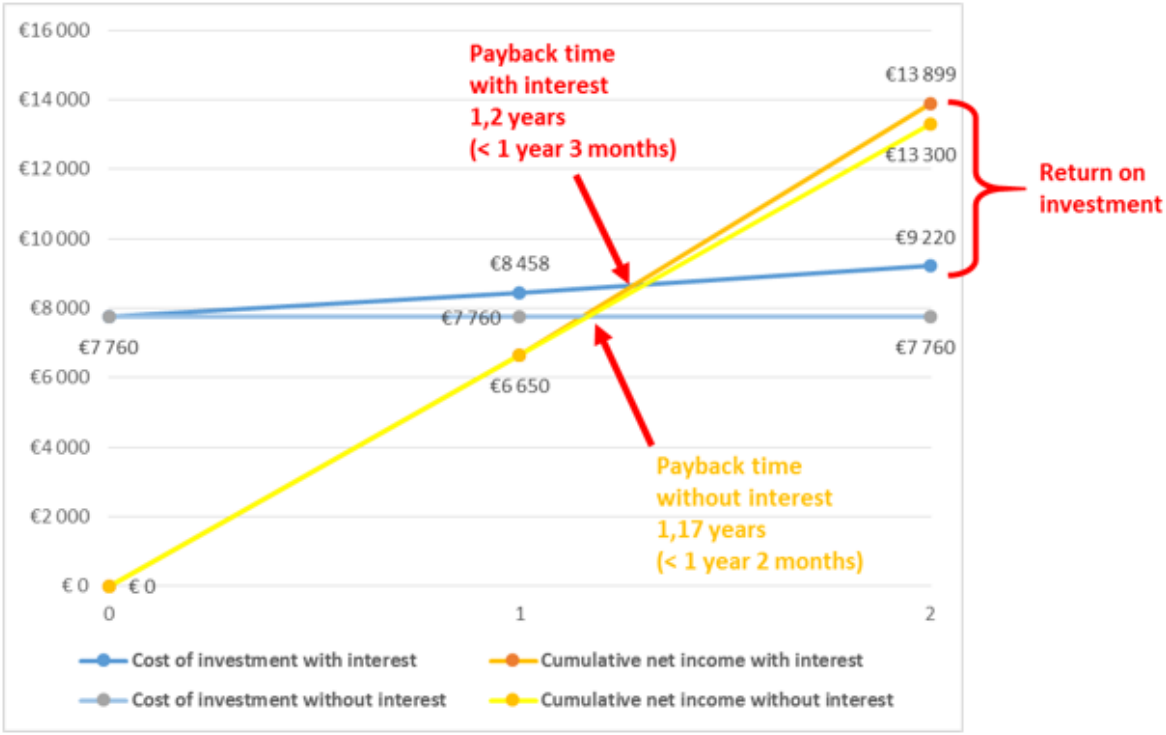
Table 3. Values of profitability calculations.

PAYBACK	ARR	NPV	IRR
1 year 2 months	36 %	3 938 €	45 %

The previous calculations do not consider the rate of interest. It is expected that the money invested today is more valuable now than after a certain amount of time. In this example case, it is expected that the investment will generate 6 650 € on each year of

its economic life. With an interest rate of 9,0 %, the net income of the first year is 6 650 € and for the second year 7 249 € (1,09 * 6 650 €). The interest also needs to be paid for the money invested until the money has been paid back. At the time of purchase the cost of the investment was 7 760 €. At the end of its first year the cost has generated interest and the new value is 8 458 € (1,09 * 7 760 €). At the end of the second year of the investment, the new value of the cost with interest is 9 219 € (1,09 * 8 458 €).

When taking the interest into account, there is a new estimate for the payback time. The interest-bearing payback time is where the two curves, cost curve with interest and cumulative net income curve with interest, meet each other (Picture 10). The payback time with interest affected is 1,2 years, so the money invested is expected to be retrieved in less than 1 year and 3 months. The estimate without interest was 1 year and 2 months. The money that is received after payback time is the return on investment also known as the profit.



	At the end of year	0	1	2
	Cost of investment with interest	€7 760	€8 458	€9 220
	Cumulative net income with interest	€0	€6 650	€13 899
	Cost of investment without interest	€7 760	€7 760	€7 760
	Cumulative net income without interest	€0	€6 650	€13 300

Picture 10. Payback time and profit with and without interest.

Based on the example calculations, the investment is profitable. It is expected to be paid back in less than one year and three months. The economic life of the investment is two years, so during the last 9 months of its useful life the investment is expected to generate profit. The return on investment is expected to be over 5 000 €. The interest-free account rate of return is 36 %, net present value 3 938 € and internal rate of return 45 %.

5 CONCLUSION

The purpose of using Screen Personas is to generate added-value through simplifying business processes, enhancing user motivation, increasing user productivity and improving efficiency. The case implemented at Wärtsilä studied how Screen Personas would fit the Marine Solutions' business processes and how cost-efficient it would be to build simplified user interfaces, automate steps and default values.

As nearly unknown by its capabilities and possibilities, Screen Personas was studied as a development project. Warranty handlers at Wärtsilä had a need for a simpler user interface to manage the warranty claims more effectively. The case was ideal as it was broad and consisted of many different process steps. To not only examine Screen Personas on the surface, its functionalities and capabilities were also analysed extensively due to the description of the case.

There were multiple successes during the development of the user interface. Having a basic knowledge of JavaScript was enough to get started with the scripting tool. Writing and recording scripts was easy, and what was delightful was the speed of the script execution. It was expected that it would take much time to render complicated scripts. However, complicated scripts were created and all of them executed in a short time. The scripting functionality ended up being the most value-adding feature.

Other valuable features included the possibility to hide unnecessary site content and the theming option. Tabs, buttons and fields that were not needed in the warranty handling process were made invisible without having an effect on the response time. Also, the ability to create consistent look and feel across the warranty handling screens was beneficial. The theming option enabled the look and feel to be defined in one screen and then applying it to other screens without having to do it manually with every single screen.

Despite the advantages of Screen Personas, there were some functionalities that restricted making the most of the interface. Tables could not be moved as it would have had a significant impact on the response time. Error messages caused problems, as depending on the values entered during claim processing, there is a variation between the amount and differency of error messages and pop-ups based on the value inputs. Also, browsers limited the performance because the popular Google Chrome browser was not entirely compatible with Screen Personas.

Despite the disadvantages, the final result of the Screen Personas user interface was profitable. Even though the information from one process step could not be merged into one screen, there were other features that generated added-value and brought the benefits. Due to the limitations of Screen Personas, the profitability of the new user interface was overestimated. The risk to exceed in the costs was accurately estimated, but it was not estimated that the expected benefits would not be completely realised. Due to the 10-minute change in the estimate of saved time, the profitability decreased noticeably, but was anyway considered as a profitable and beneficial investment.

Once the new warranty handling user interface has been taken into use, the warranty claims can be processed more efficiently in the future. With the new interface, the warranty handlers process claims with fewer screens, tabs, clicks and data inputs, the excessive navigation has been reduced, unnecessary information has been hidden and some steps and values have been automated. The benefits of these outcomes include saved time, increased efficiency, decreased amount of data errors and improved data quality and consistency.

To conclude, Screen Personas is a cost-efficient and profitable tool to be used in simplifying SAP screens. Its scripting functionality can bring many value-adding benefits that improve the efficiency and quality. It can ease the work of the users, enhance the user experience and most of all improve user motivation.

REFERENCES

Bhimani, A., Horngren, C., Datar, S. & Rajan, M. 2012. Management and cost accounting. 5th edition. Harlow: Prentice Hall/Financial Times.

Braun, K., Tietz, W. & Harrison, W. 2010. Managerial Accounting. 2nd edition. USA New Jersey: Pearson Education.

Eclipse 2012. Concept: Use-Case Model. Referred 14 August 2017 http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/concepts/use_case_model_CD178AF9.html

Eklund, I. & Kekkonen, H. 2014. Kannattavuuslaskenta ja hinnoittelu. Helsinki: Sanoma Pro Oy.

Filenius, M. 2015. Digitaalinen asiakaskokemus. Jyväskylä: Docendo Oy.

Hietikko, E. 2008. Tuotekehitystoiminta. 1st edition. Kuopio: Savonia-ammattikorkeakoulun kuntayhtymä.

Hyysalo, S. 2009. Käyttäjä tuotekehityksessä. Tieto, tutkimus, menetelmät. Helsinki. Available at https://shop.aalto.fi/media/attachments/a9bf5/Kayttaja%20tuotekehityksessa.pdf

Investopedia 2017. Residual value. Referred 27 August 2017 http://www.investopedia.com/terms/r/residual-value.asp

Jokela, T. 2010. Navigoi oikein käytettävyyden vesillä.

Jormakka, R., Koivusalo, K., Lappalainen, J. & Niskanen, M. 2015. Laskentatoimi. Helsinki: Edita Publishing Oy.

Lean Lion 2017. Miksi 5S? Referred 14 August 2017 https://www.leanlion.com/miksi-5s/

Lean Manufacturing Tools 2017. What is 5S; Seiri, Seiton, Seiso, Seiketsu, Shitsuke. Referred 21 July 2017 http://leanmanufacturingtools.org/192/what-is-5s-seiri-seiton-seiso-seiketsu-shitsuke/

Oulasvirta, A. 2011. Ihmisen ja tietokoneen vuorovaikutus. Helsinki: Raamatutrükikoda.

Ross, J. 2011. Why Don't Usability Problems Get Fixed? Referred 21 April 2017 http://www.uxmatters.com/mt/archives/2011/02/why-dont-usability-problems-get-fixed.php

Saariluoma, P. 2004. Käyttäjäpsykologia. 1st edition. Vantaa: WSOY.

SAP 2016. SAP Screen Personas 3.0 Performance Optimization Guide. Referred 20 March 2017 https://wiki.scn.sap.com/wiki/display/Img/Personas+3.0+Performance+Optimization+Guide

SAP 2017. Technology Platform / SAP Screen Personas. Referred 20 March 2017 https://www.sap.com/product/technology-platform/screen-personas.html#

The App Solutions 2016. Why the Cost of App Development Can Differ 10 Times? Referred 28 August 2017 https://theappsolutions.com/blog/marketing/app-development-cost-differ/

Tutorialspoint 2017. SAP ERP Introduction. Referred 20 March 2017 https://www.tutorialspoint.com/sap/sap_introduction.htm

Usability.gov 2017. Scenarios. Referred 14 August 2017 https://www.usability.gov/how-to-and-tools/methods/scenarios.html

UX 2013. User tasks vs. user goals? Referred 14 August 2017 https://ux.stackexchange.com/questions/42398/user-tasks-vs-user-goals

Wiio, A. 2004. Käyttäjäystävällisen sovelluksen suunnittelu. 1st edition. Helsinki: Edita Publishing Oy.

Yalantis 2017. How much does it cost to design an app? Referred 26 August 2017 https://yalantis.com/blog/how-much-does-it-cost-to-design-an-app/